





CIRCUIT FOR AND METHOD OF CONTROLLING SUBROUTINE

Patent number: JP57143642
Publication date: 1982-09-04
Inventor: DONGUSANGU ROBAATO KIMU
Applicant: BURROUGHS CORP
Classification:
- **international:** **G06F9/42; G06F9/40;** (IPC1-7): G06F9/42; G06F9/46
- **european:** G06F9/42
Application number: JP19820006720 19820118
Priority number(s): US19810231553 19810204

Also published as:

 E P0057312 (A2)
 US 4459659 (A1)
 E P0057312 (A3)
 E P0057312 (B1)

Report a data error here

Abstract not available for JP57143642

Abstract of correspondent: **US4459659**

Subroutine control apparatus for providing shared subroutine control for a plurality of executing tasks. Multiple levels of subroutine entry are provided for each task by employing a plurality of selectably accessible stacks, one for each task, along with corresponding pointer registers. These provide storage for a plurality of return addresses as required for each task during task performance. In addition, an updatable significantly faster access register is provided for each task for storing its most recent return address so as to permit return addresses to be rapidly made available when an executing task reaches the end of a subroutine.

Data supplied from the **esp@cenet** database - Worldwide

PATENT ABSTRACT OF JAPAN

(11)Publication number: 57-143642

(43)Date of publication of application: 04.09.1982

(21)Application number: 57-006720

(71)Applicant: Burroughs Corporation (Detroit, MI)

(22)Date of filing: 18.01.1982

(72)Inventor: Kim; Dongsung R. (Laguna Hills, CA)

(54)Subroutine control apparatus and subroutine control method

(57) Subroutine control apparatus for providing shared subroutine control for a plurality of executing tasks. Multiple levels of subroutine entry are provided for each task by employing a plurality of selectably accessible stacks, one for each task, along with corresponding pointer registers. These provide storage for a plurality of return addresses as required for each task during task performance. In addition, an updatable significantly faster access register is provided for each task for storing its most recent return address so as to permit return addresses to be rapidly made available when an executing task reaches the end of a subroutine.

⑫ 公開特許公報 (A)

昭57—143642

⑤Int. Cl.³
G 06 F 9/42
9/46

識別記号 庁内整理番号
6745—5B
6745—5B

⑬公開 昭和57年(1982)9月4日

発明の数 3
審査請求 未請求

(全 14 頁)

⑭サブルーチン制御回路およびサブルーチン制御方法

アメリカ合衆国カリフォルニア
州ラグーナ・ヒルズ・グリソン
・ロード24872

⑮特 願 昭57—6720

⑯出 願 人 パロース・コーポレーション

⑰出 願 昭57(1982)1月18日

アメリカ合衆国ミシガン州デト

優先権主張 ⑱1981年2月4日⑲米国(US)

ロイト・パロース・プレイス

⑳231553

(番地なし)

㉑発 明 者 ドングサング・ロバート・キム

㉒代 理 人 弁理士 深見久郎 外2名

明 細 書

1. 発明の名称

サブルーチン制御回路およびサブルーチン制御方法

2. 特許請求の範囲

(1) 1以上のサブルーチンを用いてデータ処理動作を実行するためにデータ処理システムに用いるサブルーチン制御回路であって、

複数のサブルーチンリターンアドレスを記憶するための第1のリターンアドレスストレージ手段と、

最新のリターンアドレスを記憶するための第2のリターンアドレスストレージ手段とを備え、前記第2のリターンアドレスストレージ手段は、前記第1のリターンアドレスストレージ手段よりも意味のある程度に速いアクセスを提供し、

前記システムによって与えられたサブルーチンエントリ指示にตอบสนองして、指示されたサブルーチンに対するリターンアドレスを決定し、かつ、このリターンアドレスを前記第1および第2のリタ

ーンアドレスストレージ手段の両方に記憶する第1の制御手段と、

前記システムによって与えられたサブルーチンリターン指示にตอบสนองして、前記第2のリターンアドレスストレージ手段に含まれた最新のリターンアドレスがアクセスされるようにし、かつ、前記システムに転送されるようにする第2の制御手段と、

前記サブルーチンリターン指示にตอบสนองして、前記最新のリターンアドレスが前記システムに転送された後に、次の最新のリターンアドレスが自動的に前記第1のリターンアドレスストレージ手段からアクセスされ、かつ前記第2のリターンアドレスストレージ手段に記憶されるようにする第3の制御手段とを備える、サブルーチン制御回路。

(2) 前記第1のリターンアドレスストレージ手段が、複数のリターンアドレスを記憶するためのスタックを備え、

スタック中に記憶されたリターンアドレスの番号を指示するスタックレベルポインタを記憶する

ためのスタックレベルポインタレジスタが設けられ、

前記第2のリターンアドレスストレージ手段が高速アクセスレジスタを含み、

前記第1の制御手段が、前記スタックレベルポインタによって指示された前記スタック中におけるあるレベルにリターンアドレスを記憶するために動作し、

前記第1および第2の制御手段が、スタックレベルポインタを正確に更新された状態に維持するために動作し、および、

前記第3の制御手段が、前記スタックレベルポインタによって指示されたレベルにおける前記スタックからの次の最新のリターンアドレスをアクセスする、特許請求の範囲第1項記載のサブルーチン制御回路。

(3) 前記システムが複数のタスクを実行し、各タスクは1以上のサブルーチンを含んでいてもよく、

それぞれの第1および第2のリターンアドレス

- 3 -

求の範囲第1項、第2項または第3項記載のサブルーチン制御回路。

(5) 前記第1のリターンアドレスストレージ手段が空である時を決定する手段を備える、特許請求の範囲第4項記載のサブルーチン制御回路。

(6) 同時の動作を提供するために、前記第1、第2および第3の制御手段と協働する手段を含む、特許請求の範囲第1項または第2項記載のサブルーチン制御回路。

(7) データ処理システムに用いるサブルーチン制御回路であって、複数のタスクを実行し、タスクは1以上のサブルーチンを含んでいてもよく、

各々のタスクの実行中に、各々のタスクに対するリターンアドレスを記憶するための、各タスクごとの第1のリターンアドレスストレージ手段と、

各々のタスクの最新のリターンアドレスを記憶するための、各タスクごとの第2のリターンアドレスストレージ手段とを備え、前記第2のリター

- 5 -

ンストレージ手段が各タスクごとに設けられており、

前記エントリおよびリターン指示が、該当するタスクの識別により完成され、

前記第1の制御手段が、各決定されたリターンアドレスを、識別されたタスクのそれぞれの第1および第2のリターンアドレスストレージ手段に記憶し、

前記第2の制御手段が、前記システムへの転送のために、識別されたタスクの第2のリターンアドレスストレージ手段から最新のリターンアドレスをアクセスし、および、

前記第3の制御手段によって、次の最新のアドレスが、識別されたタスクの第2のリターンアドレスストレージ手段に記憶するため識別されたタスクの第1のリターンアドレスストレージ手段からアクセスされる、特許請求の範囲第1項または第2項記載のサブルーチン制御回路。

(4) 第1のリターンアドレスストレージ手段がもはやリターンアドレスを記憶することができない時を決定するための手段を備える、特許請求

- 4 -

ンアドレスストレージ手段は、前記第1のリターンアドレス手段よりも意味のある程度に速いアクセスを提供し、

タスクがサブルーチンの初めに達するときに前記システムによって供給されたタスク識別指示およびサブルーチンエントリに回答して、指示されたサブルーチンに対するリターンアドレスを決定し、かつ、このリターンアドレスをタスクのそれぞれの第1および第2のリターンアドレスレジスタの両方に記憶する第1の制御手段と、

タスクがサブルーチンの終りに達するときに前記システムによって供給されたタスク識別指示およびサブルーチンエントリに回答して、識別されたタスクのそれぞれの第2のリターンアドレスストレージ手段に記憶された最新のリターンアドレスがアクセスされかつ前記システムに転送されるようにする第2の制御手段と、

前記サブルーチンリターン指示に回答して、次の最新のリターンアドレスが、識別されたタスクのそれぞれの第1のリターンアドレスストレージ

- 6 -

手段からアクセスされ、かつ、識別されたタスクのそれぞれの第2のリターンアドレスストレージ手段に記憶されるようにする第3の制御手段とを備える、サブルーチン制御回路。

(8) 各タスクの第1のリターンアドレスストレージ手段が、複数のリターンアドレスを記憶するためのスタックを含み、

各タスクごとに、それぞれのスタック中に記憶されたリターンアドレスの番号を指示するスタックレベルポインタを記憶するスタックレベルポインタレジスタが設けられており、

各タスクの第2のリターンアドレスストレージ手段が高速アクセスレジスタを含み、

前記第1の制御手段が、リターンアドレスを、それぞれのスタックレベルポインタによって指示されたあるレベルでスタックに記憶するために動作し、

前記第1および第2の制御手段が、スタックレベルポインタを正確に更新された状態に維持するために前記エントリおよびリターン指示にตอบสนองし

- 7 -

データ処理システムにおけるサブルーチン制御方法であって、タスクは1以上のサブルーチンを含んでいてもよく、

実行中のタスクがサブルーチンの初めに達するときにタスク識別指示およびサブルーチンエントリを提供し、

実行中のタスクがサブルーチンの終りに達するときにタスク識別指示およびサブルーチンリターンを提供し、

サブルーチンエントリ指示にตอบสนองしてリターンアドレスを決定し、かつ、このリターンアドレスを識別されたタスクに対応する第1のリターンアドレスストレージ手段に記憶し、

各タスクごとに最新の計算されたリターンアドレスを、識別されたタスクに対応する第2のリターンアドレスストレージ手段に記憶し、前記第2のリターンアドレスストレージ手段は、前記第1のリターンアドレスストレージ手段よりも意味のある程度に速いアクセスを提供し、および、

サブルーチンリターン指示にตอบสนองして、識別さ

- 9 -

て動作し、および、

前記第3の制御手段が、それぞれのスタックレベルポインタによって指示されたレベルでスタックからの次の最新のリターンアドレスをアクセスする、特許請求の範囲第7項記載のサブルーチン制御回路。

(9) 前記サブルーチン制御手段が、前記システムにより与えられたタスク識別指示にตอบสนองして、前記第1の制御手段によってリターンアドレスを、識別されたタスクに対応するスタックの適当なレベルに記憶するのに使用するために、識別されたタスクに対するそれぞれのスタックレベルポインタをそれぞれのレベルポインタレジスタからアクセスする手段を含む、特許請求の範囲第8項記載のサブルーチン制御回路。

(10) 同時の動作を提供するために、前記第1、第2および第3の制御手段と協働する手段を含む、特許請求の範囲第7項、第8項または第9項記載のサブルーチン制御回路。

(11) 複数のタスクの実行により動作する

- 8 -

れたタスクの第2のリターンアドレスストレージ手段中に記憶された最新のリターンアドレスを前記システムに転送するためにアクセスし、かつ、その後、識別されたタスクの第1のストレージ手段からの次の最新のリターンアドレスで置換える、サブルーチン制御方法。

3. 発明の詳細な説明

共に願渡されたアメリカ合衆国特許出願である、1980年5月6日に出願されたアメリカ合衆国特許出願連続番号第147,149号で、発明者がD. R. KimおよびJ. H. Mc Clintockである「マイクロ命令タスキングを使用するパイプラインされマイクロプログラムされたディジタルデータプロセッサ(Pipelined Microprogrammed Digital Data Processor Employing Microinstruction Tasking)」、および、1980年5月6日に出願されたアメリカ合衆国特許出願連続番号第147,251号で、発明者がD. R. KimおよびJ. H. Mc Clintockである「マイクロ命令レベルでのタスキングを使用

- 10 -

するマイクロプログラムされたディジタルデータ処理システム (Microprogrammed Digital Data Processing System Employing Tasking at a Microinstruction Level) には、この出願に関係する内容が含まれている。

本願出願人の同時に出願され、共に譲渡された出願である1981年2月4日に出願されたアメリカ合衆国特許出願連続番号第231,554号(本願と同日付で出願した特許出願に対応)の「多相サブルーチン制御回路 (Multi-Phase Subroutine Control Circuitry)」もまたこの出願に係る。

この発明は、一般的に、ディジタルデータ処理システムにおけるデータ処理動作を実行するための改善された手段と方法に関し、特に、マルチプログラミングおよびマルチプロセッシング環境におけるサブルーチン動作の制御のための改善された手段と方法に関する。

前述した特許出願には、マイクロプログラムされたデータ処理システムの実施例が開示されてお

- 1 1 -

この発明の好ましい実施例においては、複数の選択的にアクセス可能なスタックを、各タスクごとに1つずつ、対応するポインタレジスタと一緒に使用することにより、特に有利なサブルーチン制御能力が与えられている。また、リターンアドレスが必要なときに即座に利用できるようにするために、各タスクの最新のリターンアドレスを記憶するための比較的高速のアクセスレジスタが設けられている。動作は、サブルーチンエントリの多数のレベルが各タスクごとに調節されるのを可能にし、一方また、多くのタスクが同一のサブルーチンを共有するのを可能にするような方法で行なわれる。さらに、サブルーチン制御動作は、マルチプログラミングおよびマルチプロセッシング環境において、複数の同時に実行しているタスクに対して多相サブルーチン制御を提供できるようにするために行なわれる。

この発明の特有の性質は、他の目的、特徴、利点および用途はもちろんのこと、添付図面とともに以下の発明の詳細な説明から明らかとなろう。

- 1 3 -

り、それは、マイクロ命令レベルでのタスクのマイクロプログラミングおよびマイクロプロセッシングの両方を利用するような方法でタスクを実行する。この出願の1つの目的は、この種のシステムにおけるサブルーチン動作を制御するための改善された手段と方法を提供することである。

この発明のより一般的な目的は、データ処理システムにおける改善されたサブルーチン制御手段および方法を提供することである。

この発明の付加的な目的は、サブルーチンエントリの多くのレベルを調節することができ、かつ、多数の命令および/またはタスクの中でサブルーチンを共用する、データ処理システムにおける多相サブルーチン制御回路を提供することである。

特定の好ましい実施例においては、この発明は、前述した特許出願中に開示されたタイプのデータ処理システムに組合せて用いるようにされており、そこでは、マイクロ命令レベルでのマルチプログラミングおよびマルチプロセッシングを提供するような方法で複数のタスクが同時に実行される。

- 1 2 -

図面の記号中同一番号および同一文字は同一要素を表わす。

この記載の目的のために、この発明のサブルーチン制御回路の好ましい実施例が、前述した特許出願中に開示されているタイプのデータ処理システムに特別に組合せて用いるようにされている実施例のために記載されている。しかしながら、この発明によって提供されるサブルーチン制御能力はまた、他のタイプのシステムにおける使用のために実施され得るということを理解すべきである。

ここに紹介するサブルーチン制御回路の好ましい実施例の詳細な説明をよりよく理解するために、前述した特許出願中に開示されたシステムの適当な部分の簡単な説明をまず初めに行なう。これらの特許出願の完全な開示はここに組み入れられているものとして考へるべきである。

前述した特許出願中に開示されたデータ処理システムの好ましい実施例においては、高レベル入力命令が、実行されるべき複数のタスクとして観察される。各タスクは、1以上のタスクマイクロ

- 1 4 -

命令に分割される。各タスクマイクロ命令の実行は、3ステージ(リード、計算およびライト)によって行なわれ、各ステージは1つのクロックを必要とする。マルチプログラミングは、各タスクマイクロ命令が属するタスクにかかわらずに混合した仕方です。タスクマイクロ命令を実行することにより達成される。マルチ処理は、お互いに120°位相を異にして動作し同一の物理的なハードウェアを共有する3つの独立したプロセッサとして事実上動作するような方法でパイプラインされた3ステージのアーキテクチャを使用することにより達成される。各クロック期間中、各プロセッサは、異なるタスクマイクロ命令に対応するリード、計算およびライトステージのうちの異なるものを実行する。換言すれば、各クロック期間中、リード動作は第1のタスクマイクロ命令に対して実行されることが可能であり、計算動作は第2のタスクマイクロ命令に対して実行されることが可能であり、ライト動作は第3のタスクマイクロ命令に対して実行されることが可能である。

- 15 -

ることとを含む。

この次に続く計算ステージにおいては、選択されたマイクロ命令が実行される間に計算動作は実行される。また、(リードステージの間の条件選択データ読出しによって決定された)選択された条件は、次のマイクロ命令データを作るのに使用される。

次に続くライトステージにおいては、これはタスクマイクロ命令の実行における最後のステージであるが、ライト動作は、マイクロ命令動作の結果がストレージに書込まれる間に実行される。また、計算ステージの間に作られた次のマイクロ命令データは、次のマイクロ命令を選択するのに使用される。

第2図は、前述した特許出願において開示されたシステムが、どのように、マルチプログラミングとマルチプロセッシングの両方を利用する方法でタスクマイクロ命令を実行するかを示す例である。第2図におけるこの例は、次の3つの計算の同時動作を示す。 $(A+B) + (C+D) = H$ 、

- 17 -

第1図および第2図は、前述した特許出願において開示されたシステムの動作例を示す。第1図は基本的に、3つの120°位相を異にしたプロセッサがどのように3つのタスクマイクロ命令 T_x 、 T_y および T_z を同時に実行するかを示す。第1図における文字R、CおよびWは、それぞれ、各タスクマイクロ命令の実行中に実行された3つの連続的なリード、ライトおよび計算ステージを指す。前述した出願のシステムにおけるタスクマイクロ命令の典型的な3ステージ動作は以下のとおりである。

第1のステージの間、リード動作は、特定の選択されたタスクマイクロ命令の実行を準備するためになされる。これは、マイクロ命令実行の間に使用されるべき適当なオペランドデータをストレージから読出すことと、次のマイクロ命令アドレスを決定するのに用いる条件選択データを読出すことと、次に続く計算ステージの間データバス機能を制御するのに用いる制御信号を抽出するためマイクロ命令の適当なフィールドをデコードす

- 16 -

$(A+B) - E = I$ 、 $(C+D) - E = J$ 。 T_A から T_J までの10個のタスクは次のように特徴づけられる。

タスク	オペレータ
$T_A = T_A \#$ 、 $W T_A \#$	VALC A
$T_B = T_B \#$ 、 $W T_B \#$	VALC B
$T_C = T_C \#$ 、 $W T_C \#$	VALC C
$T_D = T_D \#$ 、 $W T_D \#$	VALC D
$T_E = T_E \#$ 、 $W T_E \#$	VALC E
$T_F = T_F \#$	ADD (A+B) = F
$T_G = T_G \#$	ADD (C+D) = G
$T_H = T_H \#$	SUBTRACT F - G = H
$T_I = T_I \#$	SUBTRACT F - E = I
$T_J = T_J \#$	SUBTRACT G - E = J

前記例に対しては、“オペランドフェッチ”タ

- 18 -

スク $T_A - T_E$ の各々は、少なくとも3クロック持ち時間（上記において“W”によって示されている）をその間にもつ2つのタスクマイクロ命令を必要とすることを仮定している。“ADD”および“SUBTRACT”タスク $T_A - T_J$ の各々は、1クロック期間（上記および第2図において下付文字を有さない“B”によって示されている）のみを必要とすることもまた仮定している。また第2図において、タスクマイクロ命令が実行されないとき“NOP”マイクロ命令が示されているのがわかる。

前述した出願中に開示されたデータ処理システムの好ましい実施例の上記の簡単な説明から、計算動作の間に、特定のタスクマイクロ命令が実行されかつ次のマイクロ命令データもまた作られるということが思い起されるであろう。この発明の好ましい実施例は、このタスクに対して実行されるべき次のタスクマイクロ命令がサブルーチンの最初のタスクマイクロ命令であるということを、この次のタスクマイクロ命令データが指示してい

- 19 -

サブルーチンが入れられるべきときにシステムによって作られたエントリ番号Eによって、第3図のサブルーチン制御回路がリターンアドレスRAを形成するため記号で表わされたオフセット値Fを現在のマイクロ命令アドレスPAに加算する。この計算されたリターンアドレスRAは、その後、エントリ番号Eを作ったタスクに対応するスタック中に記憶され（各タスクごとに1つのそのようなスタックがある）、そして、スタックレベルポインタは1だけインクリメントされかつ各々のレジスタ中にセーブされる。典型的には、各スタックは、たとえば、15のリターンアドレスが各タスクに対して対応するスタック中に記憶されるようにするため、サブルーチンエントリの15のレベルを調節する。

各スタックにおける最新のリターンアドレスが、対応する高速アクセスレジスタ中に別々に記憶されるようにするために準備がなされ、そのため最新のリターンアドレスは、サブルーチンリターンが要求されるとき、次のマイクロ命令アドレスと

- 21 -

る状態に関する。そのような状態では、次のマイクロ命令データは、サブルーチンの第1のマイクロ命令のアドレスを与えるだけでなく、記号で表わされたオフセット値Fとともにサブルーチンエントリ番号Eをもまた与える。この記号で表わされたオフセット値Fは、サブルーチンに対するリターンマイクロ命令アドレスが記号で表わされたオフセット値Fを現在のマイクロ命令のアドレスPAに加算することにより得られるような値をもつ。サブルーチンの最後のマイクロ命令の計算ステージの間、システムは、適当なリターンアドレスが次のマイクロ命令アドレスとしての使用のために与えられることをリクエストするリターン番号Rを与える。

第3図は、前述した特許出願のシステム中に応用されたマルチレベルサブルーチン制御回路の好ましい実施例を示す。この実施例を詳細に説明する前に、詳細な説明がより容易に理解されるようにするため、まず初めに、全体の動作の機能上の説明をする。

- 20 -

して使用するため迅速にシステムへ供給され得る。サブルーチンリターンは、タスク実行の計算ステージの間にシステムがリターン番号Rを供給するとき開始される、ということが思い出されるであろう。このリターン番号によって、タスクに対する別々に記憶された最新のリターンアドレスがシステムに送られ、かつ、タスクに対する次の最新のリターンアドレスによって別々のストレージ中に置き換えられる。また、タスクに対して対応するスタックポインタは、この最新のリターンアドレスがシステムに転送されたということを反映するために1だけデクリメントされる。

第3図に示した好ましい実施例の上記に要約したサブルーチン制御動作は、3つのステージにおいて実行される。そのステージは、第1図および第2図に示したようなシステムのマルチプログラミングおよびマルチプロセッシング能力に寄与するのはもちろんこれを利用するために、前述した特許出願中に開示されたシステムの前述した3ステージ リード、計算およびライト動作と同期し

- 22 -

て動作する。特に、第3図に示したこの好ましいサブルーチン制御回路は、お互いに120°位相を異にして動作する3つの別々のサブルーチン制御回路を事実上提供するような方法で構成かつ配置されており、各回路は異なるタスクに対するサブルーチン制御を提供する。

次に第4図および第5図の例を参照する。第4図は、たとえば4つのサブルーチンコールSUB-1、SUB-2、SUB-3およびSUB-4を含むタスクTを示す。第5図は、タスクの実行中の特定の時間 t_0 - t_1 におけるタスクTに対するリターンアドレススタック20および高速アクセスレジスタ22の内容を示す。

第4図に示すように、タスクTは、主要部および4つのサブルーチンコールSUB-1、SUB-2、SUB-3およびSUB-4からなる。タスクTの初めにおいては(時間 t_0)、スタック20および高速アクセスリターンアドレスレジスタ22は空であり、スタックポインタPはレベル0を指示する。時間 t_0 においてSUB-1が見

- 23 -

されかつスタック20のレベル2および高速アクセスレジスタ22に置かれるようにする。これは、第5図の t_0 において示されている。また、ポインタPは、1だけインクリメントされレベル3を指示する。SUB-3が時間 t_1 において完了すると(第4図)、対応するリターンアドレスRTN-3は、高速アクセスレジスタ22から即座にアクセスされかつスタック20からの次の最新のリターンアドレスRTN-2によってとって代わられる。これは第5図の時間 t_1 において示されている。第5図の t_1 において、リターンが実行されるので、ポインタPがレベル2を指示するため1だけデクリメントされることもわかる。リターンアドレスRTN-3がいまだにレベル2に留まっていることが理解できるであろう。しかし、もしレベル2が他のリターンアドレスを受取ればRTN-3は書き直されるので、このことは全く重要でない。したがって、明確のために、第5図における t_1 に対してレベル2が空であることが示されている。この約束は、第5図を通じて使用

- 25 -

出されると、SUB-1に対して対応するリターンアドレスRTN-1は計算されかつスタック20および高速アクセスレジスタ22中に記憶される(第5図)。スタックポインタPは、レベル1が次の利用できるスタック位置であることを示すためにレベル1を指示するため1だけインクリメントされる。

第4図に示すように、SUB-1の実行の間に時間 t_2 においてSUB-2が見出される。第5図において t_2 に対して示したように、SUB-2に対するリターンアドレスRTN-2は、計算されかつスタック20中のレベル1に記憶され、そしてポインタPはレベル2を指示するため1だけインクリメントされる。高速アクセスレジスタ22においてRTN-2はまたRTN-1にとって代わる。つまり、RTN-2が今や最新のリターンアドレスだからである。

SUB-2の実行中、第3のサブルーチンSUB-3は、時間 t_3 において見出され、そしてそれは、第3のリターンアドレスRTN-3が計算

- 24 -

されている。

第4図に示すように、 t_4 におけるSUB-3の完了後、SUB-2が続きかつ t_5 で完了する。高速アクセスレジスタ22中の対応するリターンアドレスRTN-2は、このようにアクセスされ、次の最新のリターンアドレスRTN-1によって置き換えられ、かつポインタPはレベル1を指示するため1だけデクリメントされる。これは、第5図の t_5 において示されている。

時間 t_6 において、SUB-4が見出され(第4図)、第5図における t_6 に対して示すように、これは、対応する計算されたリターンアドレスRTN-4がスタック20中のレベル1に置かれ、かつ高速アクセスレジスタ22中に置かれるようにする。そして、ポインタPはレベル2を指示するためインクリメントされる。 t_7 においてSUB-4が完了すると(第4図)、リターンアドレスRTN-4は、高速アクセスレジスタ22からアクセスされ、かつ次の最新のリターンアドレスRTN-1によってとって代わられる。これは、

- 26 -

第5図の1, において示されている。一方、ポインタPはレベル1を指示するため1だけデクリメントされる。

第4図に示すように、その後、SUB-1が続き、それは1, において完了する。そのとき、RTN-1は高速アクセスレジスタ22からアクセスされ、かつ、処理はタスクTの主要部に戻る。したがって、第5図における1, に対して示したように、スタック20および高速アクセスレジスタ22はいまや空であり、かつポインタPはレベル0を指示している。

先に指摘したように、第3図に示した好ましい実施例によって提供されたサブルーチン制御動作は、お互いに120°位相を異にして動作する3つの別々のサブルーチン制御回路を事実上提供するような方法で実現されており、各回路は異なったタスクに対するサブルーチン制御を与えている。したがって、第3図の好ましい実施例において、第4図および第5図に関連して上述した実例となるタスクTに対する動作は、3つまでの異なった

- 27 -

な動作は第6図に示されている。第6図は、第1図と概略同様のタイプの図であるが、それに加えて、各タスクごとにサブルーチンエントリおよびサブルーチンリターンの特定の事例中に生じている対応するサブルーチン制御ステージS-1, S-2およびS-3を示している。

特に、第6図は、第1図と同様に、タスクTx, TyおよびTzが、それぞれプロセッサ#1, #2および#3により、お互いに120°位相を異にして同時に実行されているのを示す。この発明の好ましい実施例をここに記述するために、一例として次のことを仮定する。すなわち第6図において、タスクTx, TyおよびTzの最初に示されたタスクマイクロ命令Tx#, Ty#およびTz#の各々は、その計算(C)動作中サブルーチンエントリ信号を供給しており、それぞれのタスクに対する次のタスクマイクロ命令(Txs, Tys, あるいはTzs)が特定のサブルーチンの最初のマイクロ命令であるべきことを示している。したがって、各々のマイクロ命令Tx#,

- 29 -

タスクに対して、お互いに120°位相を異にして同時に実行されるということを理解すべきである。これは、サブルーチンエントリあるいは特定のタスクに対するサブルーチンリターンにตอบสนองして必要とされるサブルーチン制御動作の実行に対する3つの連続的なステージS-1, S-2およびS-3(ステージごとに1クロック)を提供することにより達成される。さらに、単一のクロック期間中において、各クロック期間中ステージS-1, S-2およびS-3の各々が異なったタスクに対するそれぞれのサブルーチン動作を実行するような方法で、3つのステージS-1, S-2およびS-3のすべてを実行するために準備がなされる。

これらのサブルーチン制御ステージS-1, S-2およびS-3は、それぞれ、前述した特許出願のシステムの計算、ライトおよびリードステージに同期してそれぞれの動作を行なうこともまた注意すべきである。このシステムは第1図および第2図に関連して簡単に記載された。そのよう

- 28 -

Ty#, およびTz#, の計算ステージ(C)の間、ステージS-1, S-2およびS-3からなるサブルーチン動作は、対応する計算動作中システムによって供給されたサブルーチンエントリ信号Eにตอบสนองして、各タスクごとに始められる。結果として、各タスクごとの対応するリターンアドレスは、計算され、かつそのそれぞれのスタック20および高速アクセスレジスタ22中に記憶される。そして、第4図および第5図に関して前述したように、それぞれのスタックレベルポインタPはインクリメントされる(たとえば時間t, を注目されたい)。その後、各サブルーチンの最後のタスクマイクロ命令中(第6図においてタスクマイクロ命令Tx\$, Ty\$, およびTz\$として示されている)、ステージS-1, S-2およびS-3からなるサブルーチン動作は、対応する計算動作中システムによって与えられたサブルーチンリターン信号Rにตอบสนองして再び始められ、それによって、それぞれの高速アクセスレジスタ22中の対応するリターンアドレスが、それ

- 30 -

のそれぞれのタスクに対する次のマイクロ命令アドレスとしての使用のためシステムに送られ、一方また、対応するスタックレベルポインタPがデクリメントされ、かつ次の最新のリターンアドレス（もし与えられれば）が高速アクセスレジスタ22中に記憶される（たとえば第4図および第5図における時間t₁を注目されたい）。

もちろん、第6図において例示の目的で示されているサブルーチンエントリおよびサブルーチンリターン動作は、各プロセッサごとに独立して生じ、図示されたような特定の時間に生ずる必要はない、ということが理解されるべきである。しかしながら、第6図に示した特定の時間は、サブルーチン制御動作が、どのようにしてここに記述した好ましい実施例中の3つまでのタスクに対して同時に実行され、かつシステムの対応する計算、ライトおよびリード動作に同期して典型的に実行されるかを説明するのに有益である。

上記一般かつ機能的な説明を考慮すれば、第3図に示した好ましいサブルーチン制御回路は、

- 3 1 -

ントリが指示されると、サブルーチン信号は、タスク番号T_i（これはまたステージS-3にも加えられる。）、現アドレスPA_i、記憶化されたオフセット値F_i、ポインタP_i（これはT_iに回答してステージS-3により与えられる。）により指示された次の利用可能なスタックレベルおよびサブルーチンエントリ信号E_iを示す。

もしサブルーチンリターンが指示されると、与えられたサブルーチン信号は、タスク番号T_i、ポインタP_i（ステージS-3により与えられる）によって指示された次の利用可能なスタックレベルおよびサブルーチンリターンR_iだけを示す。後でステージS-3の記載のところで検討するけれども、ステージS-3に与えられたタスク番号T_iによって、そのタスクに対する最新のリターンアドレスがマルチプレクサ32の入力に加えられる。サブルーチンリターンのS-1の間にマルチプレクサ32に印加されたリターン信号R_iによって、その後、このリターン信号がこのタスクに対する次のマイクロ命令アドレスとして使用す

- 3 3 -

ステージS-1、S-2およびS-3の各々に与えられた構造および動作を記述することにより詳細に検討できるであろう。このことについては、好ましい実施例において、各ステージは1クロック期間中に実行され、単一のクロック期間中に各ステージが異なるタスクに対するその動作を実行する、ということをお願いすべきである。たとえば、第6図におけるクロック期間4に注目すれば、その間、S-3動作がタスクT_xに対して実行され、S-2動作がタスクT_yに対して実行され、S-1動作がタスクT_zに対して実行される。

ステージS-1

第3図に示したように、ステージS-1は、S-1レジスタ30、マルチプレクサ32およびロジック34を含む。S-1レジスタ30は、サブルーチンエントリあるいはサブルーチンリターン位置のいずれかが指示されるとき、タスクマイクロ命令の計算動作の間に与えられたサブルーチン信号を記憶するために働く。もしサブルーチンエ

- 3 2 -

るためシステムに送られる。

ステージS-1はまたロジック34を含み、これにはE_i、R_iおよびP_iが加えられる。E_iが存在していれば、ロジック34は、それぞれのスタックレベルポインタP_iがその最高のレベルであるかどうかを決定する。もしそうであれば、オーバフロー信号OVが作られ、これは、タスクT_iに対するそれぞれのスタックが一杯であり他のリターンアドレスを受入れることができないことを示す。R_iが存在していれば、ロジック34は、それぞれのスタックレベルポインタP_iがその最低のレベルにあるかどうかを決定する。もしそうであれば、信号EOが作られ、これは、T_iに対するそれぞれのスタックが空であることを示す。システムはその後、OVあるいはEO信号に回答して適当な動作を行なう。

ステージS-2

ステージS-2は、S-2レジスタ40、加算器42、インクリメント/デクリメント44およびマルチプレクサ46を含む。第3図において、

- 3 4 -

ステージS-2に与えられた信号の下付文字は、“1”から“2”に変えられることがわかるであろう。これは、先のクロック期間中にステージS-1に加えられたこれらの信号が、今、次のクロック期間においてステージS-2に加えられていることを示すためになされる。これらの下付文字“2”はまた、S-2に加えられたこれらの信号を下付文字“1”信号の新しい組と区別するために働く。その下付文字“1”の信号は、同時に、同一のクロック期間中興なったタスクに対するステージS-1に加えられる。このことは第6図に関して先に説明したとおりである（たとえば第6図のクロック期間3を参照されたい）。同一の規約は、S-3に加えられた信号に関しても用いられる。

サブルーチンエントリのステージS-2の間、それぞれのタスクに対するリターンアドレスRA_iは、加算器42により現アドレスPA_iをオフセット値F_iに加算することにより計算される。この計算されたリターンアドレスRA_iは、その

- 35 -

レジスタファイル54、ORゲート56およびマルチプレクサ58を含む。リターンアドレススタックストレージ50は、ランダムアクセスメモリを含み、このランダムアクセスメモリは各タスクに対して1つのストレージを複数のスタックに与える。これらのスタックの1つが第5図においてスタック20によって示されている。高速アクセスレジスタファイル52は、即座にアクセス可能なストレージを複数の高速アクセスレジスタに与え、これらのレジスタのうちの1つは第5図においてレジスタ22で示されている。スタックポインタレジスタファイル54は、また各タスクごとに1つずつ、ストレージを複数のスタックレベルポインタレジスタに与える。

リターンアドレススタックストレージ50は、（サブルーチンエントリのための）書込みのためあるいは（サブルーチンリタンのための）読出しのための各クロック期間中にイネーブルされ得る。したがって、E_iはストレージ50に対するライトイネーブルとして働き、かつR_iはリード

- 37 -

後、それぞれのタスク番号T_iおよびエントリ信号E_iとともにS-2レジスタ40中に記憶される。さらに、サブルーチンエントリに対して作られたエントリ信号E_iによって、インクリメント/デクリメント44がポインタP_iを1だけインクリメントし、かつまたマルチプレクサ46が一定のP_i値を出力する。これらP_iおよびP_i+1の値もまたS-2レジスタ40中に記憶される。サブルーチンリタンのステージS-2中、P_i、R_iおよびT_iのみが存在する。S-2中の動作は、次のようなものである。すなわち、R_iによりインクリメント/デクリメント44がP_iを1だけデクリメントするようにされ、またそれにより、マルチプレクサ46が結果であるP_i-1値をR_i、およびT_iとともにS-2レジスタ40中のそれぞれの位置に記憶するため進める。

ステージS-3

第3図に示したように、ステージS-3は、リターンアドレススタックストレージ50、高速アクセスレジスタファイル52、スタックポインタ

- 36 -

イネーブルとして働く。高速アクセスレジスタファイル52およびスタックポインタレジスタファイル54は、各クロック期間中興なったレジスタからの書込みおよび読出しの両者を供給できる。この能力は、ファイル52および54のために、サブルーチンステージS-1、S-2およびS-3の3つ全部が、たとえば第6図に示したように、興なったタスクに対する各クロック期間中に実行されるよう準備されている。クロック期間中読出しは常時与えられており、一方書込みはE_i、あるいはR_iのいずれかによりイネーブルされる。これは第3図において、ORゲート56のE_i+R_i出力をファイル52および54のライトイネーブル入力に加えることにより実施されている。T_iは、リターンアドレススタックストレージ50のためのスタックセクタとして働き、かつ高速レジスタファイル52およびスタックポインタレジスタファイル54の両者のためのライトレジスタセクタとして働く。T_iは、ファイル52および54の両者に対してリードレジスタセク

- 38 -

タとして働く。リターンアドレススタックストレージに対するレベルポインタは、サブルーチンエントリに対する P 、またはサブルーチンリターンに対する P 、 -1 である。

サブルーチンエントリのステージ $S-3$ の間、 E は、($S-2$ 中に計算された)リターンアドレス RA 、がタスク T 、によって選択された特定のスタック中へ、ポインタ P 、によって指示されたレベルに書込まれるようにするためにリターンアドレスストレージ 50 に対するライトイネーブルを供給する。このリターンアドレスはまた、ファイル 52 中の T 、によって選択された(たとえば第5図の時間 t 、を参照されたい)特定の高速アクセスレジスタにマルチプレクサ 58 を経て書込まれる。マルチプレクサ 58 の動作は次のようなものである。つまり、サブルーチンエントリ中(R 、が存在しないとき)、マルチプレクサ 58 はリターンアドレススタックストレージ 50 の出力よりもむしろ RA 、を通過させる。サブルーチンエントリの $S-3$ 中もまた、インクリメントさ

- 39 -

を参照されたい)。またサブルーチンリターンのステージ $S-3$ の間、デクリメントされたレベルポインタ P 、 -1 は、 T 、によって選択されたファイル 54 のスタックポインタレジスタ中に書込まれ、ストレージ 50 から次の最新のリターンアドレスの読出しを説明するためにレベルポインタ値が調整される。

T 、によって選択されたファイル 52 の特定の高速アクセスレジスタ中のリターンアドレスのステージ $S-3$ の間の書込と同時に、同一クロック期間中にステージ $S-1$ に加えられたタスク番号 T 、もまた、 T 、に対する最新のリターンアドレスの読出しのために対応する高速アクセスレジスタを選択するためにステージ $S-3$ のファイル 52 に加えられる、ということが理解されるであろう。ステージ $S-1$ に関して先に説明したように、このアクセスされたリターンアドレスは、サブルーチンリターンがステージ $S-1$ に対して指示された場合にはシステムに転送するためにステージ $S-1$ のマルチプレクサ 32 に加えられる。

- 41 -

れたレベルポインタ P 、 $+1$ は、 T 、によって選定されたファイル 54 中の特定のスタックポインタレジスタ中に書込まれる。

サブルーチンリターンのステージ $S-3$ の間、動作は主としては、リターンアドレススタックストレージ 50 に書込まれるべきリターンアドレスがないという点で、サブルーチンエントリに対して実行された動作とは異なる。その代わりとして、ストレージ 50 は R 、により読出しのためにイネーブルされる。そのような場合に、 T 、はなお、アクセスされるべきであるストレージ 50 中のスタックおよびファイル 52 中のレジスタを選択する。しかしながら、スタックレベルポインタはタスク T 、に対する次の最新のリターンアドレスを読出すために(ステージ $S-2$ 中に1だけデクリメントされた結果として) P 、 -1 の値を今はもつてであろう。そのリターンアドレスは、 T 、の対応する高速アクセスレジスタにおける書込みのためマルチプレクサ 58 を経て高速アクセスレジスタファイル 52 に進む(たとえば第6図の時間 t 、

- 40 -

T 、によって選択されたスタックポインタレジスタファイル 54 の中の特定のレジスタ中のレベルポインタ値のステージ $S-3$ の間の書込と同時に、同一クロック期間中にステージ $S-1$ に加えられたタスク番号 T 、もまた、ステージ $S-3$ のファイル 54 に加えられることが理解されるであろう。これは、同時に実行されたステージ $S-1$ 動作に対する P 、値として使用するために、タスク T 、に対するレベルポインタの現在の値を読出すために対応するスタックポインタレジスタを選択するためになされる。

以上の説明は特定の好ましい実施例を記載しているけれども、構造、配置、動作および用途における多くの変更態様がここに開示した発明の内容から外れることなく可能であるということを理解すべきである。したがって、この発明は前掲の特許請求の範囲内にあるすべての可能な変更態様を包含しているものとして考えられるべきである。

4. 図面の簡単な説明

第1図および第2図は、前述した特許出願中に

- 42 -

開示されたシステムの動作例を示す。第3図は、この発明に従ったマルチレベルサブルーチン制御回路の好ましい実施例を示す。第4図は、複数のサブルーチンコールを含む典型的なタスクタイミングの一例を示す。第5図は、第4図のタスクの実行中の第3図の実施例の基本動作を示す。第6図は、第3図のサブルーチン制御回路のステージの実行とプロセッサシステムのステージの実行との間の時間関係を示す。

図において、20はリターンアドレススタック、22は高速アクセスレジスタ、30、40はレジスタ、32、46、58はマルチプレクサ、34はロジック、42は加算器、44はインクリメント/デクリメント、50はリターンアドレススタックストレージ、52は高速アクセスレジスタファイル、54はスタックポインタレジスタファイル、56はORゲートである。

特許出願人 パロース・コーポレーション

代理人 弁理士 深見久郎

(ほか2名)



- 43 -

図面の浄書(内容に変更なし)

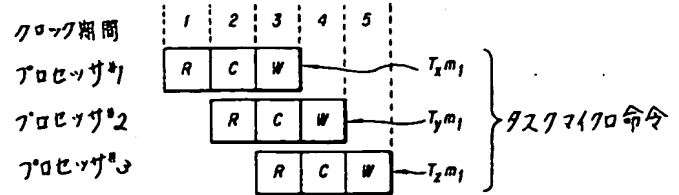


FIG. 1

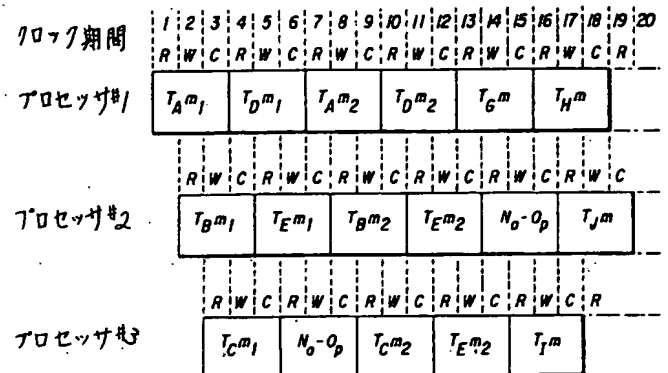
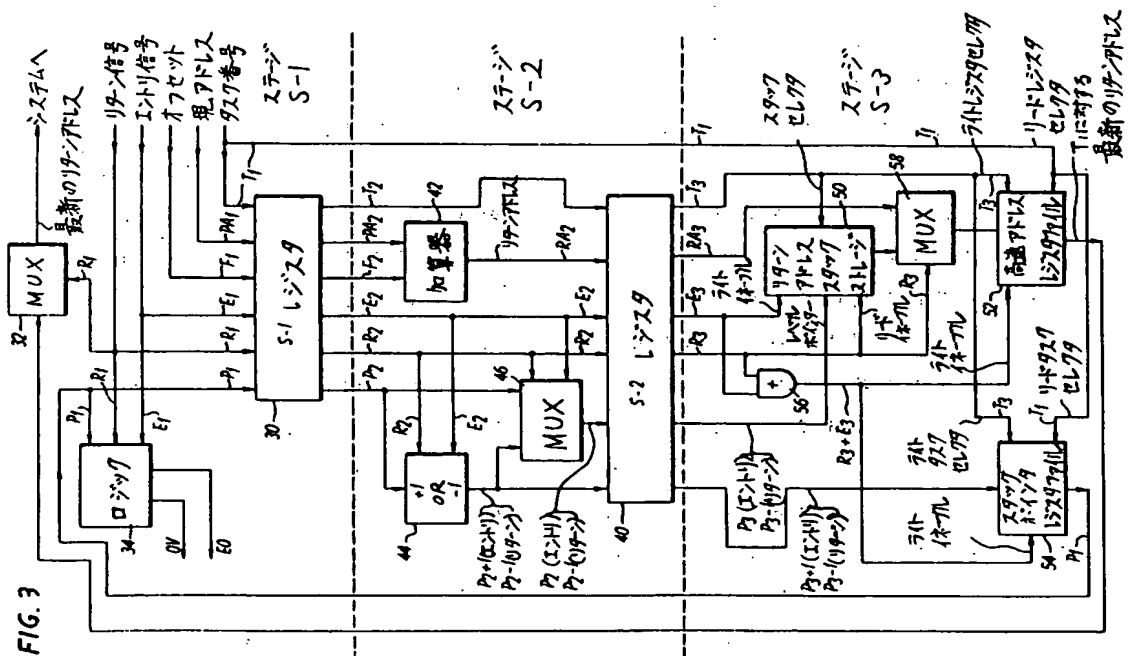


FIG. 2



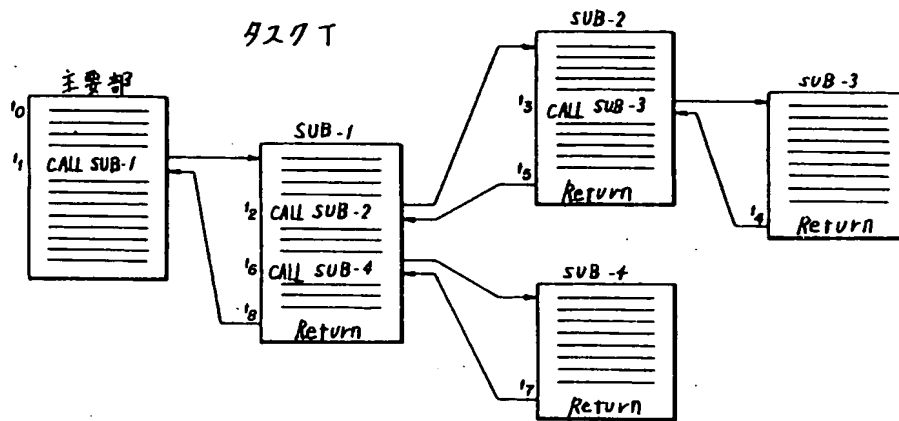


FIG. 4

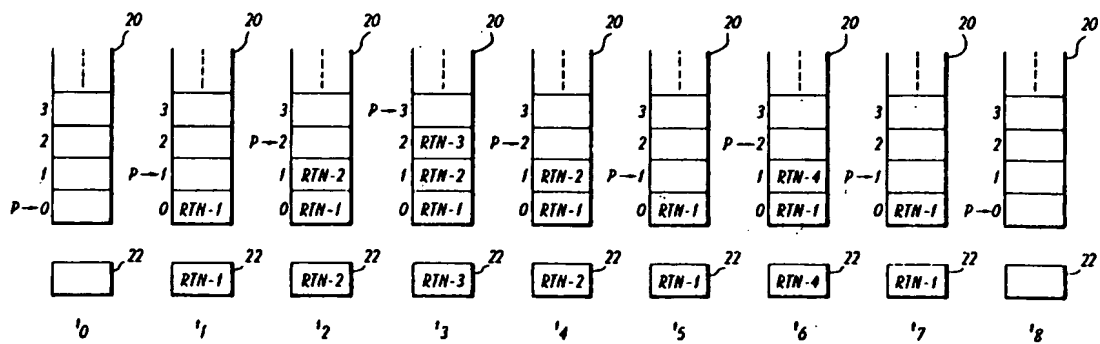
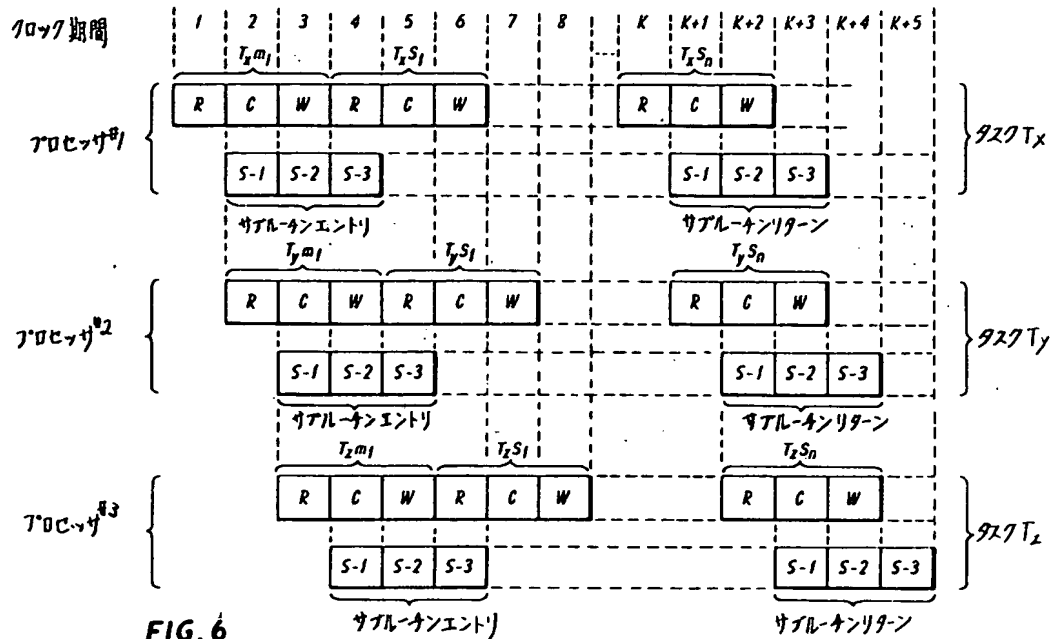


FIG. 5



手 続 補 正 書 (方 式)

昭和 57 年 3 月 4 日

特 許 庁 長 官 殿

1. 事件の表示

昭和 57 年特許願第 6720 号

2. 発明の名称

サブルーチン制御回路およびサブルーチン制御方法

3. 補正をする者

事件との関係 特許出願人

住 所 アメリカ合衆国、ミシガン州、デトロイト

パロース・ブレイス、(番地なし)

名 称 パロース・コーポレーション

代表者 ウォルター・ジェイ・ウィリアムズ

4. 代 理 人

住 所 大阪市北区天神橋 2 丁目 3 番 9 号 八千代第一ビル

電話 大阪(06)351-6239(代)

氏 名 弁理士(6474) 深 見 久 郎

5. 補正命令の日付

自発補正

6. 補正の対象

願書の 1. 発明の名称の欄および図面

7. 補正の内容

(1) 願書の 1. 発明の名称の欄に「サブルーチン
セイギョカイト
制御回路およびサブルーチンセイギョホウ
を「サブルーチン制御回路およびサブルーチン
セイギョホウ
制御方法」に訂正致します。その目的で別紙に
新たに図製した訂正願書を添付致します。

(2) 濃縮で描いた図面を別紙のとおり補充致します。なお、内容についての変更はありません。

以 上